LAB 4: DISCONTINUOUS FORCING, SIGNALS, AND NUMERICAL METHODS, PART B

1. MATLAB

MATLAB commands we use in this lab include the following.

1.1. **disp.** Displays text to the command window. For example,

```
>> disp('This is text sent to the command window')
```

See `tic` and `toc`, below.

1.2. **eulermethod_s19.** This isn't a native *MATLAB* command; download it from the labs page. This command approximates the solution to a differential equation or system using Euler's method. It takes as arguments the same arguments as we use with `ode45`[1], plus a step size to use:

```
>> eulermethod_s19(f_handle, [tmin tmax], init_cond, h);
```

1.3. **ode45.** Finds a numerical approximation to a differential equation or system of equations:

```
>> [tsol,xsol] = ode45(f_handle, [tmin tmax], init_cond);
```

Note that we can also specify, instead of just the minimum and maximum t values, a set of points at which we want to have the solution evaluated. This doesn't change the numerical calculation; it just means we know what times our `xsol` vector will be evaluated at. For example, to ensure that the solution was available at $t = 0$, $t = 0.1$, $t = 0.2$, etc., we could use

```
>> tsol = 0:.1:5;
>> [tsol,xsol] = ode45(@(t,x) [-x(2); x(1)], tsol, [0; 1]);
```

It is also possible to set options that determine how `ode45` behaves; for example, we can set the maximum step size it is allowed to try by setting up an options object and passing that to `ode45`:

```
>> options = odeset('MaxStep', 1);
>> [tsol,xsol] = ode45(@(t,x) [x(2); -x(1)], [0 10],...
          [0;1], options);
```

1.4. **ode15s.** This is another numerical solver for differential equations, and takes exactly the same arguments as `ode45`. It deals well with *stiff* systems, for which solutions have regions that change very much faster than they do in others.

1.5. **plot.** Plot one vector against another; e.g.,

```
>> plot( tesol,xesol(:,1),'-k', t45sol,x45sol(:,1),'--k' );
```

[1] Except that it doesn't support the addition of `options`.

1.6. **tic.** This starts *MATLAB*'s internal timer, so that you can see how long a command runs for; see `toc`

1.7. **toc.** This stops *MATLAB*'s internal timer, so that you can see how long a command runs for. For example, to see how long a call to `ode45` takes:

```
>> disp('timing for ode45');
>> tic
>> [t,x] = ode45(@(t,x) [1000*x(2); -1000*x(1)], [0 100],...
         [-2;5]);
>> toc
```

2. Background

In this lab we consider a circuit model, which is a second-order, linear, constant-coefficient differential equation. In the prelab we found this to be

$$y'' + 2\gamma y' + \omega_0^2 y = F(t). \tag{1}$$

The characteristic polynomial of the associated homogeneous equation is $\lambda^2 + 2\gamma\lambda + \omega_0^2$, with roots $\lambda = -\gamma \pm \sqrt{\gamma^2 - \omega_0^2}$. Thus, if γ is a small (relative to ω_0) positive number, the system is underdamped and the solution can be written in the form $y_c(t) = Re^{-\gamma t}\cos((\sqrt{\omega_0^2 - \gamma^2})t - \phi_0)$ for some R and ϕ_0. In this lab, we consider forcing functions which are discontinuous.

To solve an equation such as (1) numerically (e.g., with `ode45`), we rewrite it as a system and the numerical method then uses known data (e.g., the initial conditions and system of equations) to predict the values of the variables y and y' at a later time. In Part A we considered Euler's method to see how this works (though it is sufficiently inaccurate that it wouldn't be useful in any production context).

Note (or recall) that we define the Dirac delta function $\delta(t)$ in [BB, §5.7] to be the limit as $a \to 0$ of the function

$$\delta_a(t) = \begin{cases} \frac{1}{a}, & 0 \le t < a \\ 0, & \text{else} \end{cases}.$$

That is, it is an instantaneous impulse of unit magnitude. For mechanical systems, it induces a unit change in momentum; for electrical systems it causes a unit change in voltage or magnetic flux depending on what we're modeling.

3. Part B

In the following, we consider the equations $y'' + y' + 36y = F(t)$ and $y'' + 36y = F(t)$ for different discontinuous $F(t)$, with the goals of investigating what the differences are between the response y when $F(t)$ is a short impulse and when $F(t)$ is a delta function, and of seeing what happens when we try to cancel out a response by imposing an impulse.

Unless otherwise stated, let $I(t) = \frac{1}{a}(u_c(t) - u_{c+a})$. With the function `Impulse.m` from the course page, you can define different impulses $I_j(t)$ with

```
>>  I1 = @(t) Impulse(t,c,a1);
>>  I2 = @(t) Impulse(t,c,a2);
```

and so on (assuming that `c`, `a1`, and `a2` are already defined, of course).

Exercise 1. Review your Part A work from Exercise 4. Suppose that we consider the problem $y'' + y' + 36y = \delta(t-1)$, $y(0) = y'(0) = 0$, where $\delta(t)$ is the Dirac delta function at $t = 0$. What will the response look like? Sketch (by hand) what you think it will look like. Then solve the problem numerically with `ode15s` using $F(t) = I_j(t)$, where the $I_j(t)$ are the impulses having $c = 1$ and

$a = 0.5$, $a = 0.25$, $a = 0.05$, and $a = 0.01$. Note how your result confirms (or refutes!) your expectation. Also note that you may need an `options` setting to get `ode15s` to correctly render the solution.

Exercise 2. Now suppose that we have a circuit in which there is an (existing) undesired signal (current). This may have been started by some nonzero initial condition, say $y(0) = 0$, $y'(0) = 1$.

To make our analysis easier, let's take $\gamma = 0$ in (1). Then an initial impulse at the origin is equivalent to solving $y'' + 36y = 0$ with initial conditions $y(0) = 0$, $y'(0) = 1$. Solve this problem numerically with `ode45`, generating points corresponding to the time vector `tsol=0:.01:8` (see the *MATLAB* section above to specify the times at which to get solution values). Verify that you get the solution you expect.

Note that the signal has a period; call this T (you should be able to figure out what T is). Now suppose you want to zero out this signal by applying an impulse at $t = T$. What is the magnitude of the impulse you should apply? In what direction? Use `ode15s` to solve the equation $y'' + 36y = -k\,I(t)$ with zero initial conditions and $I(t)$ being an impulse starting at $c = T$ having width $a = 0.05$. Pick k so that you will zero out the signal (think about what the impulse does to the solution—in particular, how it is related to the initial condition on $y'(T)$). You will want to have this solution on the same points as the solution you generated with `ode45`, above. Then plot this solution added to the original signal. Does it behave as you expect? Try smaller and smaller values of a to see if, as your $-kI(t)$ converges to $-k\delta(t - T)$, the cancellation works.

Is it obvious that an impulse at $t = t_0$ is equivalent to $y(t_0) = 0$, $y'(t_0) = 1$? Solve the two problems, $y'' + 36y = \delta(t)$ with $y(0) = y'(0) = 0$ and $y'' + 36y = 0$ with $y(0) = 0$ and $y'(0) = 1$ to see—we can do this once we've covered [BB, §5.7], but even before that can use our work from Exercise 1 to see how this is the case! What's the slope of your solution at $t = 1$? How is the solution you obtained related to the solution of $y'' + y' + 36y = 0$, $y(1) = 0$, $y'(1) = 1$?

Exercise 3. Finally, let's revisit Euler's method and see what happens when we try to solve this problem with that. Generate solutions to the problem

$$y'' + 36y = F(t), y(0) = 0, y'(0) = 1$$

with `ode15s` and `eulermethod_s19`, with $F(t)$ chosen to be an impulse of width $a = 0.01$ that comes close to canceling out the signal at $t =$ one period. Plot the results together to see how they differ. How well does the numerical solutions with Euler's method do this? Can you explain what you see?

4. Reflection

Your reflection should be completed by you and your partner collaboratively, and should be about a page in length, including figures to illustrate your conclusions. In your reflection you will answer the questions given below, by following the steps:

(1) Take two minutes to think, on your own, about the answers to the following questions:
   a. Geometrically, how does Euler's method generate an approximate solution to a differential equation? How is this approximation related to the direction field for the differential equation?

   b. Given that Euler's method, and other numerical solvers, generate an approximation for the solution to a differential equation by using previous approximate values and the differential equation, what issues do numerical methods have with short impulse forcing?
   c. How can we deal numerically with problems involving delta function forces ($\delta(t)$)?
   d. Why is it hard to cancel an existing signal in a real-world circuit?

(2) Next, discuss these four questions with your partner, and, if you wish, the others at your table, by having each person of your group give their answers and reasoning. Once everyone has commented, come to a consensus as a group as to what the answers should be.
(3) With your partner, generate a reflection writeup that concisely answers the questions above.

References

[BB] Brannan, James R, and William E Boyce. *Differential Equations: an Introduction to Modern Methods And Applications.* Third edition. Hoboken, NJ: Wiley, 2015.